**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное учреждение высшего образования

**«Московский технический университет связи и информатики»**

**ФАКУЛЬТЕТ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ**

Кафедра «МКиИТ»

**Информационные технологии и**

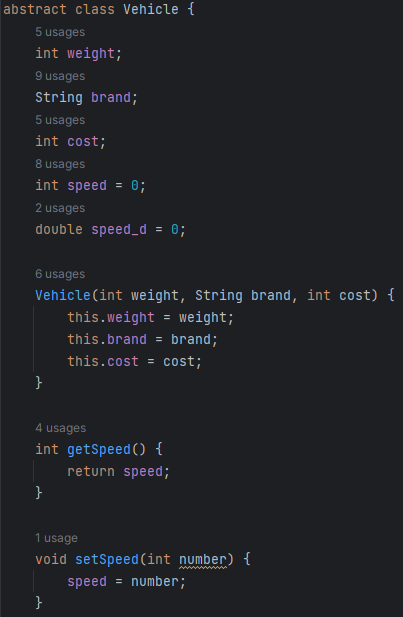
**программирование**

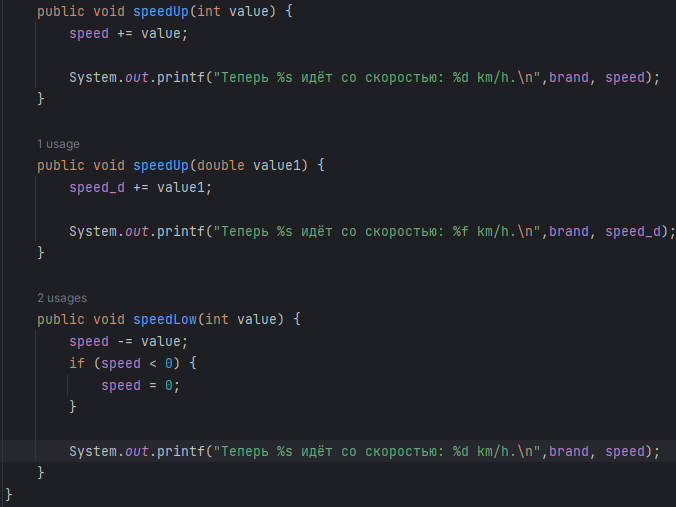
**Лабораторная работа по ИТП №2**

Выполнил: студент группы БВТ2205

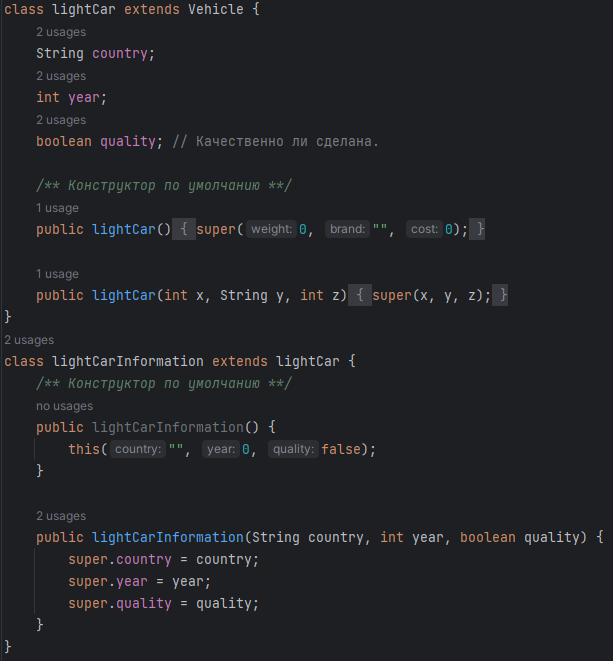
Белов Егор Олегович

1) Абстрактный класс:



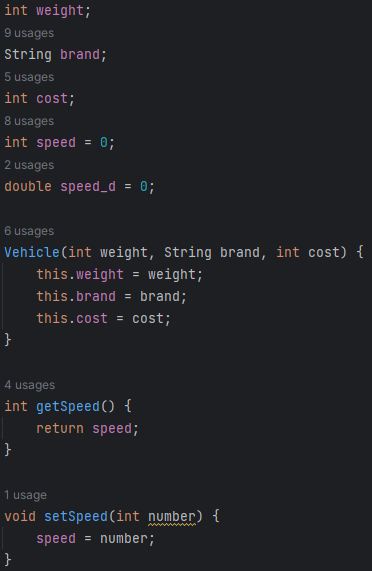


2) 2 уровня наследуемых классов с 3-мя полями и 2-мя методами, описывающих поведение объекта:

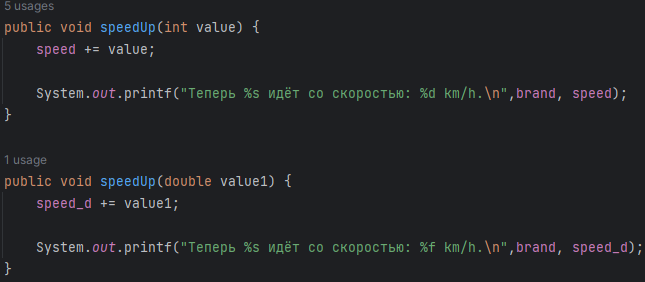


Методы находятся в суперклассе Vehicle.

3) Абстракция:

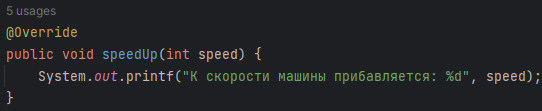


4) Перегрузка:



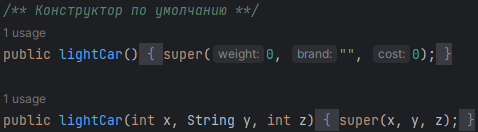
**Создал два метода с одним и тем же именем, но разными параметрами.**

**5) Переопределение:**

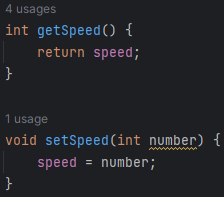
****

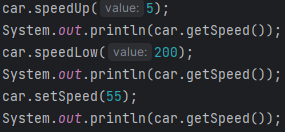
**Метод из абстрактного суперкласса.**

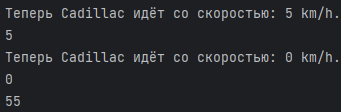
**6) Конструкторы:**

****

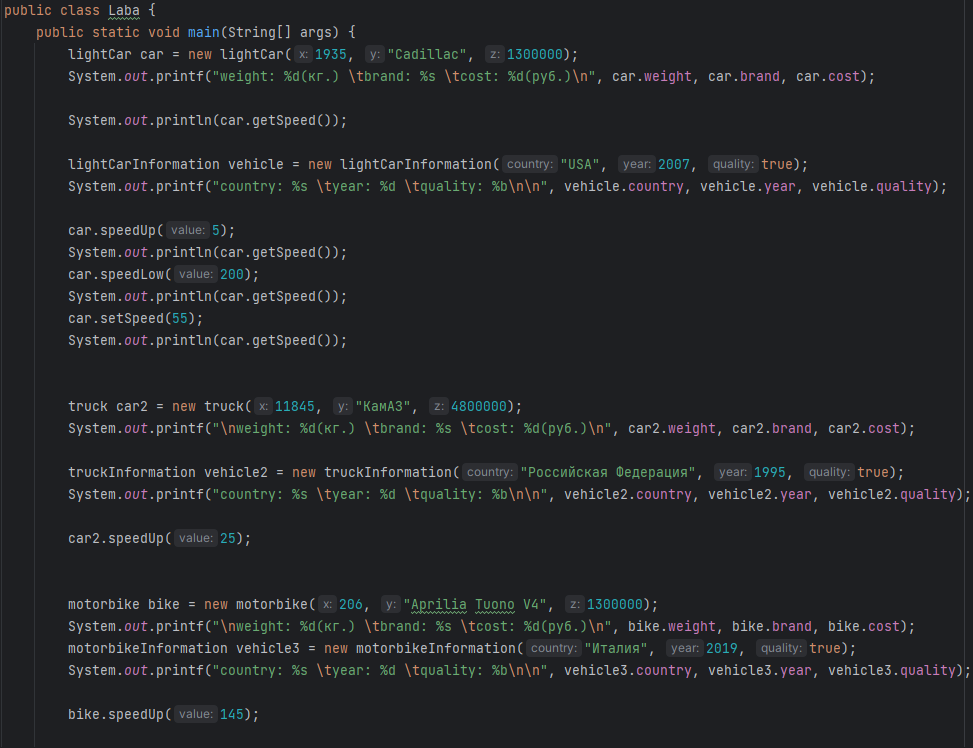
**7) Геттеры и сеттеры:**

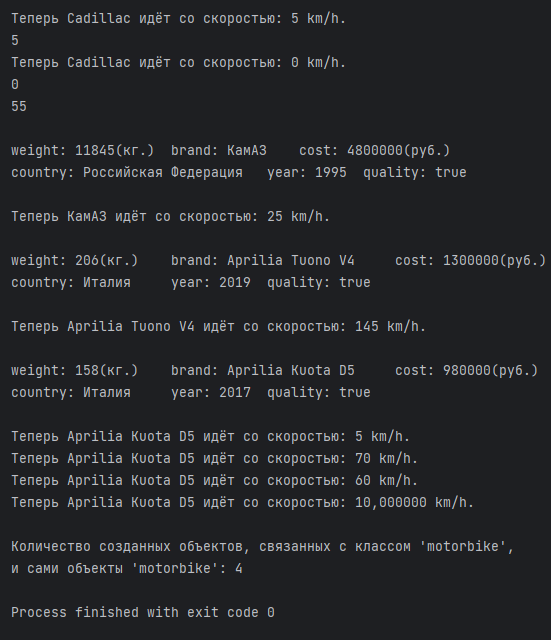
****

****

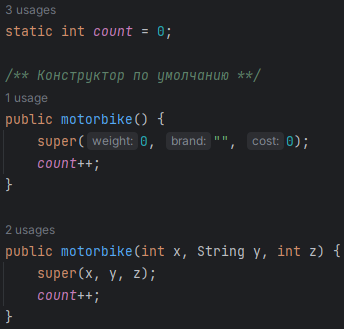
****

**8) Ввод/вывод информации:**

****

****

**9) Создание счётчика:**

****

****

****

----------------------------------------------------------------------------------------------------

Подготовить ответы на вопросы:

1. Принципы ООП, рассказать об этих принципах.
2. Можно ли наследовать от нескольких классов в Java?
3. Абстракция, переопределение, перегрузка и т.д.
4. В целом надо ориентировать по лекции в понедельник.

1) Абстракция, инкапсуляция, наследование и полиморфизм - это основные принципы объектно-ориентированного программирования (ООП), которые позволяют создавать гибкие, масштабируемые и легко поддерживаемые программы.

1. Абстракция - это процесс выделения существенных характеристик объекта и игнорирования несущественных. В Java абстракция может быть достигнута с помощью абстрактных классов и интерфейсов. Абстрактный класс - это класс, который не может быть создан напрямую, а только наследован другими классами.

2. Инкапсуляция - это механизм, который позволяет скрыть детали реализации объекта от внешнего мира и предоставить только интерфейс для взаимодействия с объектом. В Java инкапсуляция может быть достигнута с помощью модификаторов доступа (public, private, protected) и геттеров/сеттеров.

3. Наследование - это механизм, который позволяет создавать новый класс на основе существующего класса. Новый класс наследует все свойства и методы существующего класса и может добавлять свои собственные свойства и методы. В Java наследование может быть достигнуто с помощью ключевого слова **extends**.

4. Полиморфизм - это механизм, который позволяет объекту принимать несколько форм. В Java полиморфизм может быть достигнут с помощью перегрузки методов и переопределения методов.

Перегрузка методов - это создание нескольких методов с одним и тем же именем, но разными параметрами.

Переопределение методов - это создание метода в наследующем классе с тем же именем и параметрами, что и в родительском классе.

2) В Java класс может наследовать только один класс, но может реализовывать несколько интерфейсов. Это означает, что вы можете использовать наследование только для одного класса, но можете добавлять функциональность, реализуя несколько интерфейсов.

Наследование - это механизм, который позволяет классу наследовать свойства и методы другого класса, называемого суперклассом или родительским классом. Класс-наследник, также называемый подклассом или дочерним классом, может использовать свойства и методы родительского класса, а также добавлять свои собственные свойства и методы.

3)

1. Перегрузка методов - это создание нескольких методов с одним и тем же именем, но разными параметрами.

Переопределение методов - это создание метода в наследующем классе с тем же именем и параметрами, что и в родительском классе.

2. Переопределение - это механизм в объектно-ориентированном программировании, который позволяет переопределить реализацию метода в подклассе, который наследует этот метод от своего суперкласса. При переопределении метода в подклассе, метод в суперклассе становится скрытым для подкласса, и вызов метода на объекте подкласса будет вызывать переопределенную версию метода в подклассе.

Класс состоит из полей и методов.